Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

STATYSTYKA MATEMATYCZNA

z pakietem R

II. Statystyka opisowa

Przemysław Grzegorzewski
Konstancja Bobecka-Wesołowska
Marek Gągolewski

Spis treści

1 Wprowadzenie

Statystyka opisowa stawia sobie za cel stworzenie wstępnego obrazu interesującego nas zbioru danych, pochodzącego np. z badania eksperymentalnego, bądź będącego wynikiem symulacji. Metody statystyki opisowej stosuje się najczęściej jako pierwszy, a zarazem najbardziej podstawowy, etap analizy danych.

Mierzone aspekty badanych obiektów przedstawia się w postaci jednej bądź wielu zmiennych, będących wynikiem różnego rodzaju pomiarów (por. [1]). Ogólnie rzecz biorąc, rozpatrywać będziemy dwa typy czynników.

Zmienne jakościowe

Zmienna typu *jakościowego* (ang. *qualitative variable*) reprezentuje cechę obiektu występującą na jednym z kilku wyróżnionych poziomów (kategorii, klas). Na przykład zmienna „płeć" może przyjmować dwie wartości: „kobieta" bądź „mężczyzna". I tak obiekt „Jan Kowalski" najprawdopodobniej przynależy do drugiej kategorii z wymienionych.

Zmienne ilościowe

Z kolei zmienna typu *ilościowego* (ang. *quantitative variable*) opisuje mierzalną cechę obiektu, tj. wyrażalną za pomocą wartości liczbowej, np. wzrost delikwenta w centymetrach czy też błąd średniokwadratowy badanego estymatora.

Typ zmiennej determinuje możliwe do zastosowania sposoby opisu (por. [2, 3]). Poniżej przedstawiamy te najczęściej stosowane wraz ze sposobem ich uzyskania za pomocą programu `R`.

a) Zmienne jakościowe:

   i. Tabele: rozkład liczności (`table`), rozkład częstości (`prop.table`).

   ii. Metody graficzne: wykres kołowy (`pie`), wykres słupkowy (`barplot`).

b) Zmienne ilościowe:

   i. Charakterystyki liczbowe (statystyki próbkowe):
      - Charakterystyki położenia: moda, średnia (`mean`), średnia ucięta (`mean(...,trim=...)`), mediana (`median`) i inne kwantyle (`min`, `max`, `quantile`).
      - Charakterystyki rozproszenia: wariancja (`var`), odchylenie standardowe (`sd`), rozstęp międzykwartylowy (`IQR`), rozstęp (`diff(range(...))`).
      - Charakterystyki kształtu rozkładu: skośność, kurtoza.

   ii. Metody graficzne: wykres pudełkowy (`boxplot`), histogram (`hist`), wykres łodygowo-liściowy (`stem`).

Podczas niniejszych ćwiczeń zapoznamy się także z metodami ładowania zbiorów danych z plików, transformacji danych i dzielenia zbiorów na podzbiory w zależności od kategorii.

2 Zadania rozwiązane

Zadanie 2.1. Pewna grupa studentów wydziału MiNI została poproszona przez pracownicę dziekanatu o wybranie swego przedstawiciela. Kandydatami do tej zaszczytnej funkcji byli: Złotowłosa Kasia, Wąsaty Jerzy, Pulchny Stefan i Kowalska Cecylia. W głosowaniu wzięło udział 25 osób. Jesteś członkiem okolicznościowej komisji i — jako znawca programu `R` — zostałeś poproszony o wstępne zanalizowanie wyników celem opublikowania ich na internetowej stronie samorządu.

Rozwiązanie.

Pierwszą czynnością, którą należy wykonać, jest wprowadzenie danych. Dysponujemy 25 kartkami do głosowania, ich wyniki możemy zapisać w postaci wektora napisów. Używać będziemy tylko inicjałów imion.

Wprowadzanie danych

```
> glosy <- c("ZK", "ZK", "KC", "PS", "KC", "PS", "PS", "ZK", "KC", "KC",
             "PS", "KC", "KC", "WJ", "PS", "KC", "PS", "ZK", "KC", "WJ",
             "PS", "ZK", "WJ", "KC", "KC");
> glosy <- factor(glosy); # konwersja na wektor czynnikowy - wyodrębnienie klas
> length(glosy);

[1] 25
```

Są to dane typu jakościowego. Zmienna może przyjąć jedną z 4 kategorii, każda z nich odpowiada pewnej osobie, na którą można zagłosować w omawianych wyborach.

Zliczenia głosów uzyskanych przez każdego kandydata możemy dokonać za pomocą funkcji `table()`.

Tabela liczności

```
> glosyTab <- table(glosy); # tabela liczności
> print(glosyTab);

glosy
KC PS WJ ZK
10  7  3  5
```

Zwyciężyła zatem Kowalska Cecylia; otrzymała bowiem 10 głosów — najwięcej ze wszystkich kandydatów. Wyniki możemy także przedstawić w postaci *tabeli częstości*.

Tabela częstości

```
> prop.table(glosyTab);

glosy
  KC   PS   WJ   ZK
0.40 0.28 0.12 0.20
```

Kasia (nasza faworytka) otrzymała niestety tylko 20% wszystkich głosów. Zwróćmy uwagę, że funkcja `prop.table()` przyjmuje jako parametr tabelę liczności, a nie wektor `glosy`.

Jeżeli chcemy mieć dostęp oddzielnie do 4-wyrazowego wektora liczby głosów oraz do wektora odpowiadających głosom kategorii, możemy wykonać następujące polecenia.

```
> osoby <- names(glosyTab); # wektor nazw kategorii (inicjały kandydatów)
> liczbaGlosow <- as.vector(glosyTab); # liczba głosów na każdego kandydata
> print(osoby);

[1] "KC" "PS" "WJ" "ZK"


> print(liczbaGlosow);

[1] 10  7  3  5


> osoby[2]; liczbaGlosow[2]; # wyniki drugiej osoby

[1] "PS"

[1] 7
```

Za ich pomocą da się odtworzyć zawartość wektora `glosy` (z dokładnością do permutacji wyrazów).

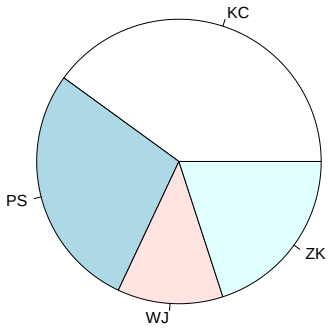
```
> rep(osoby, liczbaGlosow)
```

```
 [1] "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "PS" "PS" "PS" "PS" "PS"
[16] "PS" "PS" "WJ" "WJ" "WJ" "ZK" "ZK" "ZK" "ZK" "ZK"
```

Umiejętność przechodzenia z różnych postaci danych do innych jest istotna na przykład gdy mamy dostęp do informacji już wstępnie przetworzonych (zliczonych). W takim wypadku wprowadzilibyśmy je używając właśnie wektorów liczności oraz nazw kategorii.

W wielu zastosowaniach bardziej przyjaznymi sposobami opisu danych są wykresy. Chyba najpopularniejszym z nich jest *wykres kołowy* (ang. *pie chart*). Wykres kołowy

```
> pie(glosyTab);
```



lub równoważnie:

```
> pie(liczbaGlosow, labels=osoby);
```

Uwaga

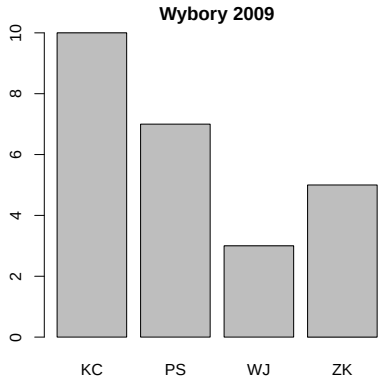
Spróbujmy także innych ustawień:

```
> pie(glosyTab, col=c("red", "blue", "yellow", "green"));
> pie(glosyTab, col=heat.colors(4));
> pie(glosyTab, border=NULL, radius=1, main="Wyniki głosowania",
    labels=c("Cecylia", "Stefan", "Jerzy", "Kasia"));
> pie(glosyTab, labels=paste(osoby, " - ", liczbaGlosow));
```

Więcej informacji na temat parametrów funkcji graficznych i sposobów ustawiania kolorów znaleźć można w *Dodatku*.

Inną metodą prezentacji danych jest *wykres słupkowy*: Wykres słupkowy

```
> barplot(glosyTab, main="Wybory 2009");
```



Uwaga

Rozważmy również:

```
> barplot(liczbaGlosow, names=osoby);
> barplot(prop.table(glosyTab), names=as.vector(prop.table(glosyTab)),
    horiz=T, legend=names(glosyTab), col=rainbow(4));
```

Z powyższych wykresów szybko możemy wywnioskować jak układają się liczby bądź proporcje głosów oddanych na poszczególnych kandydatów. Nie na darmo mówimy, że jeden obraz wart jest tysiąca słów! ⊡

Zadanie 2.2. W pliku `samochody.csv` zamieszczono historyczne dane dotyczące parametrów samochodów kilku wybranych marek.

a) Zmienna `mpg` zawiera dane odpowiadające liczbie mil, przejechanych przez dany samochód na jednym galonie paliwa. Utwórz zmienną `zp` opisującą zużycie paliwa mierzone w litrach na 100 kilometrów.

b) Poddaj nowo utworzoną zmienną kategoryzacji, tworząc następujące klasy:

Kod kategorii	Zużycie paliwa [l/100 km]
mało	mniejsze niż 7
średnio	nie mniejsze niż 7, lecz mniejsze niż 10
dużo	nie mniejsze niż 10

c) Dla otrzymanych danych utwórz i omów wykres słupkowy.

Rozwiązanie.

Baza samochodów przechowywana jest w pliku typu CSV (ang. *comma-separated values*). Jest to zwykły plik tekstowy o określonej strukturze. Można go podejrzeć z użyciem np. programu *Notatnik*:

Plik typu CSV

```
mpg;cylindry;moc;przysp;rok;waga;producent;marka;model;cena;legenda
43,1;4;48 ;21,5;78;1985;2;Volkswagen;Rabbit Dl ;2400;America=1
36,1;4;66 ;14,4;78;1800;1;Ford      ;Fiesta    ;1900;Europe=2
32,8;4;52 ;19,4;78;1985;3;Mazda     ;GLC Deluxe;2200;Japan =3
39,4;4;70 ;18,6;78;2070;3;Datsun    ;B210 GX   ;2725;
36,1;4;60 ;16,4;78;1800;3;Honda     ;Civic CVCC;2250;
19,9;8;110;15,5;78;3365;1;Oldsmobile;Cutlass   ;3300;
19,4;8;140;13,2;78;3735;1;Dodge     ;Diplomat  ;3125;
...
```

Ładowanie pliku CSV

Pierwszy wiersz pliku określa nazwy kolumn (zmiennych). W kolejnych wierszach mamy dostęp do informacji dla poszczególnych samochodów. Załadujmy tę bazę jako ramkę danych R-a. Służy do tego funkcja `read.csv()`.

```
> samochody <- read.csv("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv");
Error in read.table(file = file, header = header, sep = sep, quote = quote,  :
  more columns than column names
```

Niestety, ta operacja nie powiodła się. Jest to spowodowane tym, że domyślnie R zakłada format pliku CSV zgodny z (niepisanym) standardem:

Własność	Wartość domyślna	Parametr `read.csv`
Sposób oddzielania zmiennych	"," (przecinek)	`sep`
Separator części ułamkowej liczby	"." (kropka)	`dec`
Obecność nazw zmiennych w pierwszym wierszu	`TRUE`	`header`

W naszym przypadku kolumny są rozdzielone znakami średnika, a znakiem oddzielającym część całkowitą od ułamkowej jest przecinek. Toteż ładowanie bazy powinno się raczej odbyć w sposób następujący.

```
> samochody <- read.csv("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv",
    sep=";", dec=",");
```

Sprawdźmy, czy plik został zinterpretowany w oczekiwany sposób:

```
> class(samochody); # jest to ramka danych?
[1] "data.frame"

> head(samochody) # wyświetl kilka pierwszych wierszy
  mpg cylindry moc przysp rok waga producent      marka      model cena
1 43.1        4  48   21.5  78 1985         2 Volkswagen Rabbit Dl  2400
2 36.1        4  66   14.4  78 1800         1 Ford       Fiesta     1900
3 32.8        4  52   19.4  78 1985         3 Mazda      GLC Deluxe 2200
4 39.4        4  70   18.6  78 2070         3 Datsun     B210 GX    2725
5 36.1        4  60   16.4  78 1800         3 Honda      Civic CVCC 2250
6 19.9        8 110   15.5  78 3365         1 Oldsmobile Cutlass    3300
    legenda
1 America=1
2  Europe=2
3  Japan =3
4
5
6
```

Zauważmy, iż kolumna `legenda` nie zmieściła się na ekranie, więc została wyświetlona poniżej pozostałych.

Uwaga

Format CSV jest stosunkowo często używany do przechowywania prostych zbiorów danych. Tego typu bazy można edytować np. za pomocą arkusza kalkulacyjnego. Przy odczycie/zapisie należy wybrać tylko odpowiedni format pliku, np. w programie Excel jest to *Plik tekstowy (CSV)*, a w angielskojęzycznym OpenOffice Calc *Text CSV*.

Więcej interesujących opcji polecenia `read.csv()` znajduje się oczywiście w systemie pomocy:

```
> ?read.csv
```

Analogiczną funkcją służącą do zapisu ramek danych jest `write.csv()`.

W niniejszym zadaniu interesować nas będzie jedynie zmienna `mpg`. Jest ona typu ilościowego, jednakże poddamy ją — celem ćwiczenia — kategoryzacji.

```
> samochody$mpg
```

```
  [1] 43.1 36.1 32.8 39.4 36.1 19.9 19.4 20.2 19.2 20.5 20.2 25.1 20.5 19.4 20.6
 [16] 20.8 18.6 18.1 19.2 17.7 18.1 17.5 30.0 27.5 27.2 30.9 21.1 23.2 23.8 23.9
 [31] 20.3 17.0 21.6 16.2 31.5 29.5 21.5 19.8 22.3 20.2 20.6 17.0 17.6 16.5 18.2
 [46] 16.9 15.5 19.2 18.5 31.9 34.1 35.7 27.4 25.4 23.0 27.2 23.9 34.2 34.5 31.8
 [61] 37.3 28.4 28.8 26.8 33.5 41.5 38.1 32.1 37.2 28.0 26.4 24.3 19.1 34.3 29.8
 [76] 31.3 37.0 32.2 46.6 27.9 40.8 44.3 43.4 36.4 30.4 44.6 40.9 33.8 29.8 32.7
 [91] 23.7 35.0 23.6 32.4 27.2 26.6 25.8 23.5 30.0 39.1 39.0 35.1 32.3 37.0 37.7
[106] 34.1 34.7 34.4 29.9 33.0 34.5 33.7 32.4 32.9 31.6 28.1   NA 30.7 25.4 24.2
[121] 22.4 26.6 20.2 17.6 28.0 27.0 34.0 31.0 29.0 27.0 24.0 23.0 36.0 37.0 31.0
[136] 38.0 36.0 36.0 36.0 34.0 38.0 32.0 38.0 25.0 38.0 26.0 22.0 32.0 36.0 27.0
[151] 27.0 44.0 32.0 28.0 31.0
```

```
> length(samochody$mpg)
```

```
[1] 155
```

Wśród 155 obserwacji stwierdzamy obecność jednego braku danych. Warto usunąć ten element ciągu, gdyż w tym momencie takie oznaczenie nie wnosi niczego istotnego, a nawet czasem komplikuje użycie niektórych funkcji statystycznych; służy do tego np. funkcja `na.omit()`.

Konwersja jednostek miar

Aby dokonać konwersji jednostek mile/galon na litry/100 km, należy użyć następującego wzoru:

$$\texttt{zp} = \frac{1}{\texttt{mpg}} \frac{3{,}785 \cdot 100}{1{,}609}, \tag{1}$$

gdyż 1 mila = 1,609 km, a 1 galon = 3,785 l. Wynikowy ciąg zapiszemy jako wektor `zp`:

```
> # usuwamy braki danych:
> mpg <- na.omit(samochody$mpg);            # albo (lepiej):
> mpg <- as.vector(na.omit(samochody$mpg)); # albo:
> mpg <- samochody$mpg[!is.na(samochody$mpg)];
> # konwersja:
> zp <- 3.785*100/(mpg*1.609);
> print(zp, digits=3);                      # wypisanie
```

```
  [1]  5.46  6.52  7.17  5.97  6.52 11.82 12.13 11.65 12.25 11.48 11.65  9.37
 [13] 11.48 12.13 11.42 11.31 12.65 13.00 12.25 13.29 13.00 13.44  7.84  8.55
 [25]  8.65  7.61 11.15 10.14  9.88  9.84 11.59 13.84 10.89 14.52  7.47  7.97
 [37] 10.94 11.88 10.55 11.65 11.42 13.84 13.37 14.26 12.93 13.92 15.18 12.25
 [49] 12.72  7.37  6.90  6.59  8.59  9.26 10.23  8.65  9.84  6.88  6.82  7.40
 [61]  6.31  8.28  8.17  8.78  7.02  5.67  6.17  7.33  6.32  8.40  8.91  9.68
 [73] 12.32  6.86  7.89  7.52  6.36  7.31  5.05  8.43  5.77  5.31  5.42  6.46
 [85]  7.74  5.27  5.75  6.96  7.89  7.19  9.93  6.72  9.97  7.26  8.65  8.84
 [97]  9.12 10.01  7.84  6.02  6.03  6.70  7.28  6.36  6.24  6.90  6.78  6.84
[109]  7.87  7.13  6.82  6.98  7.26  7.15  7.44  8.37  7.66  9.26  9.72 10.50
[121]  8.84 11.65 13.37  8.40  8.71  6.92  7.59  8.11  8.71  9.80 10.23  6.53
[133]  6.36  7.59  6.19  6.53  6.53  6.53  6.92  6.19  7.35  6.19  9.41  6.19
[145]  9.05 10.69  7.35  6.53  8.71  8.71  5.35  7.35  8.40  7.59
```

Jesteśmy już gotowi do poddania każdej wartości zmiennej kategoryzacji. Wyniki umieścimy w wektorze `spalanie`. Będzie on miał taką samą długość co `zp` i każdy jego element `spalanie[i]` będzie oznaczał klasę do której wpada odpowiadający mu wyraz `zp[i]`, `i` $= 1, ..., 154$.

```
> spalanie <- rep(NA, length(zp)); # "pusty" wektor o żądanym rozmiarze
> spalanie[zp<7] <- "mało";
> spalanie[zp>=7 & zp<10] <- "średnio";
> spalanie[zp>=10] <- "dużo";
> spalanie <- factor(spalanie); # konwersja na zmienną jakościową
> print(spalanie);
```

```
  [1] mało    mało    średnio mało    mało    dużo    dużo    dużo    dużo
 [10] dużo    dużo    średnio dużo    dużo    dużo    dużo    dużo    dużo
 [19] dużo    dużo    dużo    dużo    średnio średnio średnio średnio dużo
 [28] dużo    średnio średnio dużo    dużo    dużo    dużo    średnio średnio
 [37] dużo    dużo    dużo    dużo    dużo    dużo    dużo    dużo    dużo
 [46] dużo    dużo    dużo    dużo    średnio mało    mało    średnio średnio
 [55] dużo    średnio średnio mało    mało    średnio mało    średnio średnio
 [64] średnio średnio mało    mało    średnio mało    średnio średnio średnio
 [73] dużo    mało    średnio średnio mało    średnio mało    średnio mało
 [82] mało    mało    mało    średnio mało    mało    mało    średnio średnio
 [91] średnio mało    średnio średnio średnio średnio średnio dużo    średnio
[100] mało    mało    mało    średnio mało    mało    mało    mało    mało
[109] średnio średnio mało    mało    średnio średnio średnio średnio średnio
[118] średnio średnio dużo    średnio dużo    dużo    średnio średnio mało
[127] średnio średnio średnio średnio dużo    mało    mało    średnio mało
[136] mało    mało    mało    mało    mało    średnio mało    średnio mało
[145] średnio dużo    średnio mało    średnio średnio mało    średnio średnio
[154] średnio
Levels: dużo mało średnio
```

Uzyskaliśmy tym samym to, o co nam chodziło. Warto przypomnieć, że wyrażenie

```
> spalanie[zp<7] <- "mało";
```

oznacza „weź te elementy wektora `spalanie`, które odpowiadają elementom wektora `zp`, których wartość jest mniejsza niż 7 i przypisz im kategorię `"mało"`". Kolejny raz mamy więc okazję zaobserwować, jak bardzo zwięzłe, a zarazem jak pojemne znaczeniowo są konstrukcje języka `R`.

`R` ma także wbudowaną wygodną funkcję, o nazwie `cut()`, służącą do kategoryzowania zmiennych ilościowych według podziału $(b_1, b_2], (b_2, b_3], \ldots, (b_{n-1}, b_n]$ (przedziały domknięte prawostronnie, parametr (`right=TRUE`) bądź $[b_1, b_2), [b_2, b_3), \ldots, [b_{n-1}, b_n)$ (przedziały domknięte lewostronnie, parametr (`right=FALSE`) dla pewnych $b_1 < b_2 < \cdots < b_n$.

```
> cut(zp, c(-Inf, 7, 10, Inf), right=FALSE);
```

```
  [1] [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)  [10, Inf) [10, Inf)
  [8] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [7,10)    [10, Inf) [10, Inf)
 [15] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [22] [10, Inf) [7,10)    [7,10)    [7,10)    [7,10)    [10, Inf) [10, Inf)
 [29] [7,10)    [7,10)    [10, Inf) [10, Inf) [10, Inf) [10, Inf) [7,10)
 [36] [7,10)    [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [43] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [50] [7,10)    [-Inf,7)  [-Inf,7)  [7,10)    [7,10)    [10, Inf) [7,10)
 [57] [7,10)    [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [7,10)    [7,10)
```

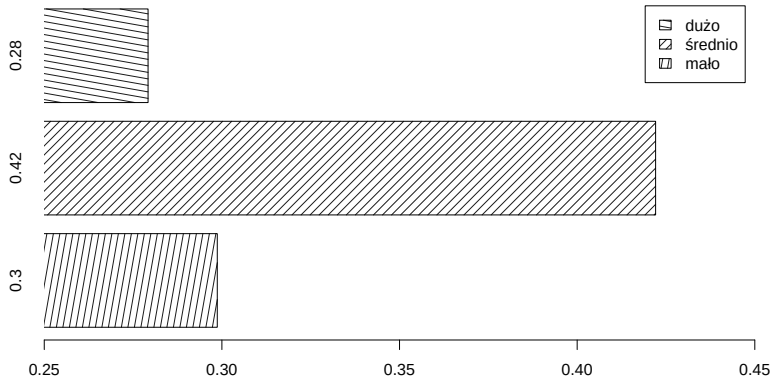
```
 [64] [7,10)    [7,10)    [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [7,10)
 [71] [7,10)    [7,10)    [10, Inf) [-Inf,7)  [7,10)    [7,10)    [-Inf,7)
 [78] [7,10)    [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)
 [85] [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [7,10)    [7,10)    [7,10)
 [92] [-Inf,7)  [7,10)    [7,10)    [7,10)    [7,10)    [7,10)    [10, Inf)
 [99] [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)
[106] [-Inf,7)  [-Inf,7)  [-Inf,7)  [7,10)    [7,10)    [-Inf,7)  [-Inf,7)
[113] [7,10)    [7,10)    [7,10)    [7,10)    [7,10)    [7,10)    [7,10)
[120] [10, Inf) [7,10)    [10, Inf) [10, Inf) [7,10)    [7,10)    [-Inf,7)
[127] [7,10)    [7,10)    [7,10)    [7,10)    [10, Inf) [-Inf,7)  [-Inf,7)
[134] [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)
[141] [7,10)    [-Inf,7)  [7,10)    [-Inf,7)  [7,10)    [10, Inf) [7,10)
[148] [-Inf,7)  [7,10)    [7,10)    [-Inf,7)  [7,10)    [7,10)    [7,10)
Levels: [-Inf,7) [7,10) [10, Inf)
```

Uzyskanym klasom przydzielimy nazwy inne niż domyślne (zob. powyżej) i narysujemy wykres słupkowy.

```
> spalanie <- cut(zp, c(-Inf, 7, 10, Inf), right=F,
    labels=c("mało", "średnio", "dużo"));
> spalTab <- table(spalanie); print(spalTab);
```

```
spalanie
   mało średnio    dużo
     46      65      43
```

```
> barplot(prop.table(spalTab), names=round(as.vector(prop.table(spalTab)), 2),
    horiz=T, legend=names(spalTab), xlim=c(0.25, 0.45), xpd=FALSE,
    col="gray10", density=15, angle=c(80,45,-10));
```



□

Zadanie 2.3. Przeprowadź wstępną analizę statystyczną zużycia paliwa (w l/100 km) samochodów opisanych w bazie `samochody.csv`.

Rozwiązanie.
Tym razem zanalizujemy wektor `zp` jako zmienną typu ilościowego. Przypomnijmy, że przechowuje on informacje o zużyciu paliwa, mierzone w litrach na 100 km, samochodów z interesującej nas historycznej bazy danych.

Charakterystyki liczbowe zmiennej ilościowej

Zacznijmy od charakterystyk liczbowych naszej próby. Zakładamy, że Czytelnik zna definicje i znaczenie poniższych statystyk. Ponadto, sugerujemy samodzielną interpretację uzyskanych wartości.

a) Charakterystyki położenia:

```
> mean(zp);     # średnia (arytmetyczna)
[1] 8.766693

> median(zp);   # mediana
[1] 8.139865

> min(zp);      # minimum
[1] 5.048053

> max(zp);      # maksimum
[1] 15.17673

> range(zp);    # min. i max. jako jeden wektor
[1]  5.048053 15.176728

> quantile(zp, c(0.1, 0.25, 0.5, 0.75, 0.9)); # kwantyle różnych rzędów
      10%       25%       50%       75%       90%
 6.190507  6.863301  8.139865 10.433264 12.296949

> summary(zp);  # wygodna funkcja, wiele statystyk
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.048   6.863   8.140   8.767  10.430  15.180

> mean(zp, trim=0.1); # średnia ucięta (po 10% obserwacji z każdej strony)
[1] 8.557733
```

b) Charakterystyki rozproszenia:

```
> var(zp);         # wariancja
[1] 5.895066

> sd(zp);          # odchylenie standardowe
[1] 2.427976

> IQR(zp);         # rozstęp międzykwartylowy
[1] 3.569962

> diff(range(zp)); # rozstęp
[1] 10.12867
```

```
> sd(zp)/mean(zp); # współczynnik zmienności

[1] 0.2769546
```

c) Charakterystyki kształtu rozkładu:

```
> library("e1071"); # musimy załadować dodatkową bibliotekę,
>                   # w niej bowiem znajdują poniższe funkcje:
> skewness(zp);     # współczynnik skośności

[1] 0.6801541


> kurtosis(zp);     # kurtoza

[1] -0.5847272
```

Warto zwrócić uwagę, że statystyki zaimplementowane w funkcjach `skewness()` i `kurtosis()` są estymatorami obciążonymi odpowiednich parametrów.

Uwaga

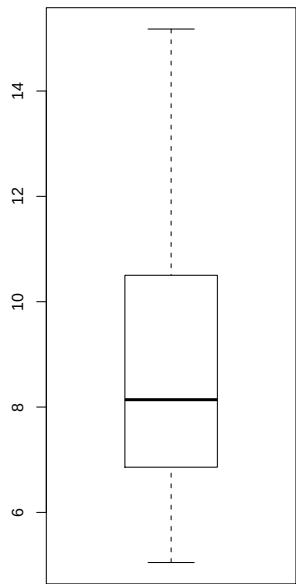
Jeżeli biblioteka `e1071` nie została zainstalowana w używanym systemie, należy to uczynić wydając komendę:

```
> install.packages("e1071");
```

Wykres pudełkowy

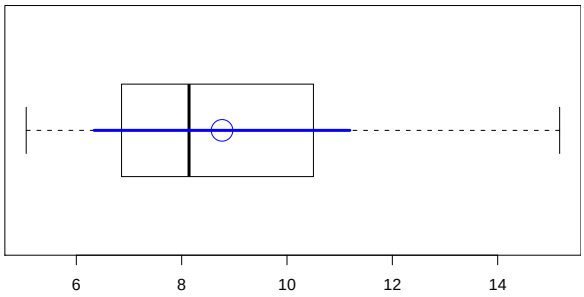
Wykres pudełkowy (ramkowy, ang. *box plot*; „pudełko z wąsami”, ang. *box-and-whisker plot*) jest wygodną metodą graficznej reprezentacji podstawowych statystyk z próby (kwartyli, minimum, maksimum) oraz identyfikacji obserwacji odstających. Domyślnie w programie `R` za „outliery” przyjmuje się obserwacje mniejsze niż $Q_1 - 1{,}5\,\mathrm{IQR}$ bądź większe niż $Q_3 + 1{,}5\,\mathrm{IQR}$, gdzie Q_1 — wartość pierwszego kwartyla z danej próby, Q_3 — trzeciego, zaś $\mathrm{IQR} = Q_3 - Q_1$.

```
> boxplot(zp);
```



Możliwe są niewielkie wariacje uzyskanego wykresu, np. narysowanie go w postaci poziomej. Dodamy doń także oznaczenie średniej z próby i odcinka średnia ± jedno odchylenie standardowe.

```
> boxplot(zp, horizontal=T);
> points(mean(zp), 1, cex=3, col="blue"); # dodanie punktu...
> lines(c(mean(zp)-sd(zp), mean(zp)+sd(zp)), c(1,1), col="blue", lwd=3); # i linii
```

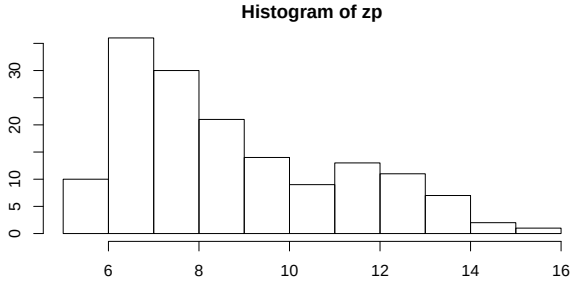


Z powyższego schematu możemy m.in. odczytać prawostronnie skośny charakter rozkładu. Nie stwierdzamy istnienia obserwacji odstających.

Histogram

Histogram jest często stosowaną graficzną formą prezentacji rozkładu empirycznego. Zaobserwowane wartości badanej zmiennej grupujemy w rozłączne klasy, będące przedziałami w $\mathbb{R}$ przeważnie o tej samej długości. Następnie zliczamy liczbę obserwacji wpadających do każdego przedziału i nanosimy wyniki na wykres podobny do słupkowego.

```
> hist(zp);
```



Dostrzegamy prawostronną skośność rozkładu. Wartości zmiennej `zp` zostały podzielone na 11 klas. `R` dokonuje automatycznego doboru tej liczby, domyślnie zgodnie z tzw. regułą Sturgesa. Inne reguły można podać za pomocą parametru `breaks=`*Identyfikator*, według wzoru:

Identyfikator	Nazwa	Wyrażenie
`"Sturges"`	Reguła Sturgesa	$k = \lceil \log_2 n + 1 \rceil$ $(n \geqslant 30)$,
`"Scott"`	Wzór Scotta dla rozkładu normalnego	$h = 3{,}5s/n^{1/3}$,
`"FD"`	Wzór Freedmana-Diaconisa	$h = 2\ \mathrm{IQR}/n^{1/3}$,

gdzie k — liczba klas, h — szerokość przedziału, n – liczba obserwacji, s — odchylenie standardowe z próby, IQR — rozstęp międzykwartylowy.

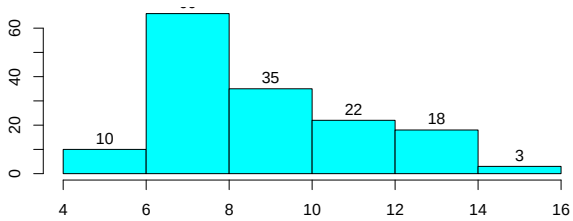
Na przykład wywołanie:

```
> hist(zp, breaks="Scott");
```

daje w wyniku liczbę klas równą 6.

Możemy także *zasugerować* (R wie lepiej, czego nam potrzeba) pożądaną liczbę klas. Warto dodatkowo w tym miejscu zanotować, iż histogram można zapisać jako obiekt; daje to sposobność odczytania jego parametrów.

```
> h <- hist(zp, breaks=5, # wygeneruj histogram z sugerowaną liczbą klas=5
                labels=T, # dodaj etykietki nad słupkami
                col="cyan", main=NA);
> h; # wyświetl parametry

$breaks
[1]  4  6  8 10 12 14 16

$counts
[1] 10 66 35 22 18  3

$intensities
[1] 0.03246753 0.21428571 0.11363636 0.07142857 0.05844156 0.00974026

$density
[1] 0.03246753 0.21428571 0.11363636 0.07142857 0.05844156 0.00974026

$mids
[1]  5  7  9 11 13 15

$xname
[1] "zp"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

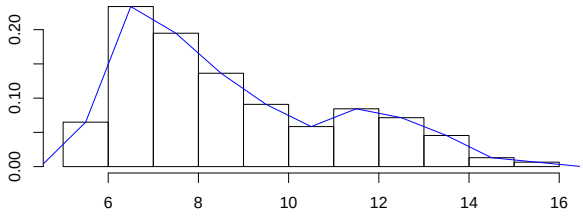


Jak łatwo zauważyć, liczba klas determinuje kształt histogramu; nie ma w tym wypadku jednej, złotej reguły dla wszystkich możliwych prób. Pożądaną kategoryzację zmiennej należy dobierać w każdym przypadku sprawdzając kilka możliwości, wybierając najbardziej informatywną i estetyczną zarazem.

Uwaga

Możemy pokusić się o narysowanie histogramu częstości (parametr `prob=TRUE`, daje to wykres, pod którym pole wynosi 1) wraz z naniesioną nań łamaną częstości. Będziemy jednak musieli ją skonstruować ręcznie, jak następuje:

```
> h <- hist(zp, prob=T, main=NA);
> szerPrzedzialu <- h$breaks[2]-h$breaks[1]; # zakładamy podział na równe części
> ileKlas <- length(h$mids); # mids - środki przedziałów
> lamanaX <- c(h$mids[1]-szerPrzedzialu, h$mids, h$mids[ileKlas]+szerPrzedzialu);
>    # potrzebne dodatkowe dwie linie zaczynające się w y=0
> lamanaY <- c(0, h$density, 0);
> lines(lamanaX, lamanaY, col="blue"); # dodaj linie
```



Nietrudno wykazać, że pole pod tak skonstruowaną łamaną częstości także wynosi 1, zatem jest ona pewnym estymatorem gęstości rozkładu.

Istnieją także inne sposoby estymowania gęstości na podstawie próby losowej, np. estymatory jądrowe. Omawiamy je pokrótce w *Dodatku*.

Wykres łodygowo-liściowy

Podobnym do histogramu jest *wykres (diagram) łodygowo-liściowy* (ang. *stem-and-leaf display*). Był on bardziej popularny w czasach, gdy komputery nie miały takich możliwości generowania grafiki, jakie mają dziś (wyświetlany jest w trybie tekstowym). Łatwo się go także rysuje na kartce.

```
> stem(zp);

  The decimal point is at the |

   5 | 033345788
   6 | 000222222334445555555556778888999999
   7 | 000122233333444444455666677889999
   8 | 0123444446666677778889
   9 | 0133447788899
  10 | 0012255799
  11 | 1344556666689
  12 | 113333679
  13 | 003444889
  14 | 35
  15 | 2
```

Dzięki niemu można wytworzyć sobie nie tylko intuicję o kształcie rozkładu, ale i także odczytać przybliżone wartości wszystkich obserwacji, np., kolejno, 5,0, 5,3, 5,3, 5,3, 5,4 itd.

Spójrzmy jak będzie wyglądał diagram tego typu po zmianie parametru skali:

```
> stem(zp, scale=2);
```

```
  The decimal point is at the |

   5 | 03334
   5 | 5788
   6 | 00022222233444
   6 | 555555556778888999999
   7 | 000122233333444444
   7 | 5566667788999
   8 | 012344444
   8 | 6666677778889
   9 | 013344
   9 | 7788899
  10 | 00122
  10 | 55799
  11 | 1344
  11 | 556666689
  12 | 113333
  12 | 679
  13 | 003444
  13 | 889
  14 | 3
  14 | 5
  15 | 2
```

□

Zadanie 2.4. Korzystając z bazy `samochody.csv`, przeprowadź wstępną analizę statystyczną zużycia paliwa (w l/100 km) oddzielnie dla samochodów produkowanych w Ameryce, Europie i Japonii. Rozważ tylko auta tańsze niż 10 000$.

Rozwiązanie.

Podział wektora na podciągi wg kategorii

Stwórzmy na nowo wektor `zp`, tym razem nie usuwając z niego braków danych. Będziemy bowiem chcieli podzielić go na rozłączne części, z których każda będzie przechowywać informacje o spalaniu samochodów z jednego z trzech regionów świata.

```
> zp <- 3.785*100/(samochody$mpg*1.609);
> samochody$producent;

  [1] 2 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 3 1 1 3 2 2 2 2 2 3 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 2 1 2 1 1 1 3 2 1 1 1 1 2 3 1 3 1 1 1 1 2
 [75] 3 3 3 3 3 1 3 2 2 2 2 3 2 3 2 3 3 2 1 3 1 1 1 1 1 3 1 3 3 3 3 3 1 1 1 2 2
[112] 3 3 3 3 2 2 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1 2 3 3 1 1 3 3 3 3 3 3 1 1 1 1 3
[149] 1 1 1 2 1 1 1
```

Znaczenie poszczególnych wartości zmiennej `producent` przechowywane jest w kolumnie `legenda` (dodanej sztucznie do bazy przez jej twórcę).

```
> head(samochody$legenda);

[1] America=1 Europe=2  Japan =3
Levels:         America=1 Europe=2 Japan =3
```

Możemy przekształcić zmienną `producent` na typ jakościowy (w tym momencie `R` traktuje ją jako zwykły wektor liczb całkowitych) i przypisać kategoriom liczbowym wartości opisowe:

```
> prod <- factor(samochody$producent);
> levels(prod) <- c("Ameryka", "Europa", "Japonia");
> head(prod);

[1] Europa  Ameryka Japonia Japonia Japonia Ameryka
Levels: Ameryka Europa Japonia


> table(prod);

prod
Ameryka  Europa Japonia
     85      26      44
```

Stwórzmy teraz trzy nowe wektory, każdy odpowiadający zużyciu paliwa aut kosztujących mniej niż 10000$ pochodzących z różnych regionów:

```
> zpA <- zp[prod == "Ameryka" & samochody$cena < 10000];
> zpE <- zp[prod == "Europa"  & samochody$cena < 10000];
> zpJ <- zp[prod == "Japonia" & samochody$cena < 10000];
```

Zwróćmy uwagę, że przywrócenie braków danych w wektorze `zp` było konieczne do podziału wartości na podzbiory; wektory `zp`, `producent` i `cena` muszą mieć tę samą długość. Pozostaje jeszcze ich powrotne usunięcie (teraz, nie wcześniej!) i sprawdzenie, czy zbiory wynikowe reprezentują to, o co zostaliśmy poproszeni.

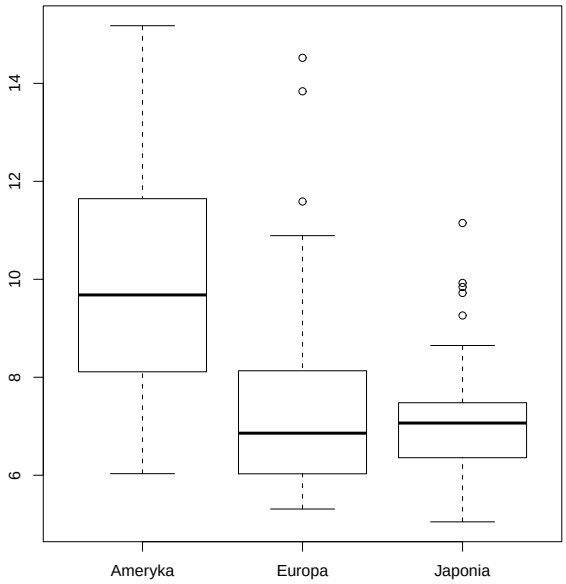
```
> zpA <- zpA[!is.na(zpA)] # usuwamy braki danych - teraz już nie są potrzebne
> zpE <- zpE[!is.na(zpE)] #
> zpJ <- zpJ[!is.na(zpJ)] #
> length(zpA) + length(zpE) + length(zpJ); # ile łącznie obserwacji?

[1] 152


> sum(samochody$cena < 10000 & !is.na(zp)) # czy tyle samo?

[1] 152
```

By uzyskać dostęp do podstawowych statystyk próbkowych, należy wywołać stosowne funkcje oddzielnie dla każdego wektora, na przykład:

```
> summary(zpA);

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.032   8.112   9.681   9.854  11.650  15.180


> summary(zpE);

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.310   6.029   6.858   7.741   8.133  14.520


> summary(zpJ);

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.048   6.358   7.065   7.215   7.462  11.150
```

Interesujące też będzie porównanie (zestawienie) rozkładów zużyć paliwa za pomocą graficznych narzędzi. W przypadku wykresu pudełkowego możemy podać na raz wszystkie wektory zawierające analizowane dane:

```
> boxplot(zpA, zpE, zpJ, names=c("Ameryka", "Europa", "Japonia"));
```



Widzimy m.in., że zużycie paliwa samochodów amerykańskich (zresztą zgodnie z obiegową opinią) jest przeciętnie większe niż innych. Jednocześnie wyniki w tej grupie charakteryzują się większą zmiennością. W próbie europejskiej i japońskiej stwierdzamy istnienie obserwacji odstających.

Uwaga

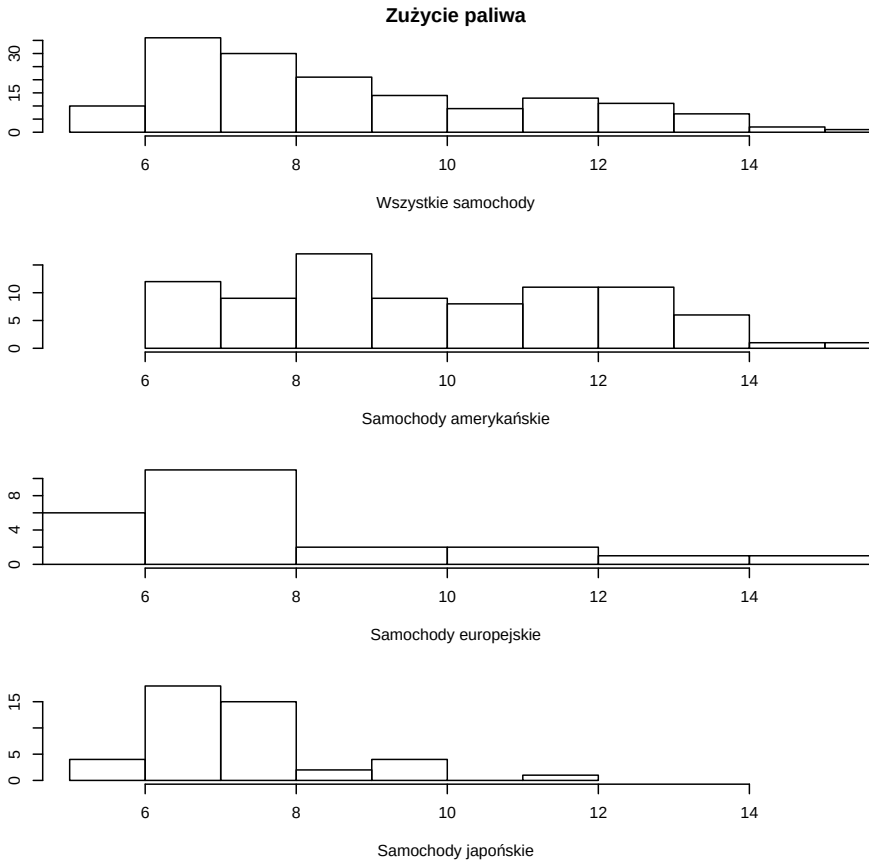
W przypadku wykresu pudełkowego istnieje inny, wygodniejszy sposób analizy zmiennej podzielonej na podzbiory. Możemy poprosić R-a, by sam to uczynił za pomocą tzw. *formuły* (więcej szczegółów podamy w części dotyczącej analizy regresji). Wynik podobny do powyższego uzyskamy za pomocą następującego kodu. Należy jednak pamiętać, by wcześniej wykluczyć nieinteresujące nas informacje (czyli auta nie tańsze niż 10 000$).

```
> zp2   <- zp[samochody$cena < 10000];
> prod2 <- prod[samochody$cena < 10000];
> boxplot(zp2 ~ prod2); # co znaczy: podziel zp2 według kategorii z prod2
```

Funkcja rysująca histogram niestety nie daje nam możliwości podania kilku zbiorów do analizy. Chcąc zatem mieć możliwość porównania rozkładów, powinniśmy umieścić je na jednym obrazku. Nadto, chcielibyśmy z pewnością umieścić również wykres dla populacji wszystkich samochodów. Wywołanie polecenia `par(mfrow=c(4,1))` nakaże R-owi utworzenie obrazka składającego z czterech podwykresów, jednego pod drugim. Powinniśmy także zapewnić, iż widoczny zakres na osi x jest taki sam w każdym przypadku.

```
> par(mfrow=c(4,1));   # 4 w 1
> zakres <- range(zp); # od min. do maks. obserwacji w całym zbiorze
> zakres

[1] NA NA
```

```
> zakres <- range(na.omit(zp)); # ach, te braki danych!
> zakres

[1]  5.048053 15.176728


> # 4 histogramy:
> hist(zp,  ylab="Liczność", xlab="Wszystkie samochody",    xlim=zakres,
     main="Zużycie paliwa");
> hist(zpA, ylab="Liczność", xlab="Samochody amerykańskie", xlim=zakres, main=NA);
> hist(zpE, ylab="Liczność", xlab="Samochody europejskie",  xlim=zakres, main=NA);
> hist(zpJ, ylab="Liczność", xlab="Samochody japońskie",    xlim=zakres, main=NA);
```



Wyciągnięcie wniosków z przedstawionych wykresów pozostawiamy Czytelnikowi. ⊡

Zadanie 2.5. Poniższe dane odpowiadają notowaniom pewnej spółki (w PLN) w kolejnych 20 dniach:

> 23.30, 24.50, 25.30, 25.30, 24.30, 24.80, 25.20, 24.50, 24.60, 24.10,
> 24.30, 26.10, 23.10, 25.50, 22.60, 24.60, 24.30, 25.40, 25.20, 26.80.

Utwórz wykres cen akcji jako funkcji czasu (*szereg czasowy*).

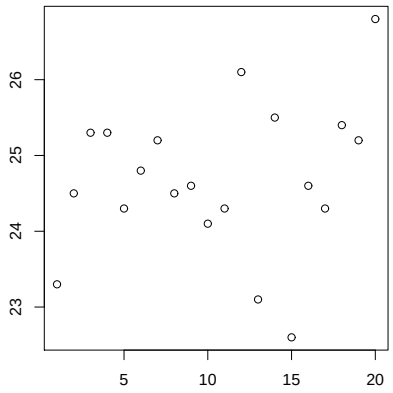
Rozwiązanie.
Nasamprzód należy wprowadzić dane do naszego ulubionego programu:

```
> cenyAkcji <- c(23.30, 24.50, 25.30, 25.30, 24.30, 24.80, 25.20, 24.50, 24.60,
  24.10, 24.30, 26.10, 23.10, 25.50, 22.60, 24.60, 24.30, 25.40, 25.20, 26.80);
```

Utworzenie wykresu dokonuje się poprzez wywołanie funkcji `plot`.

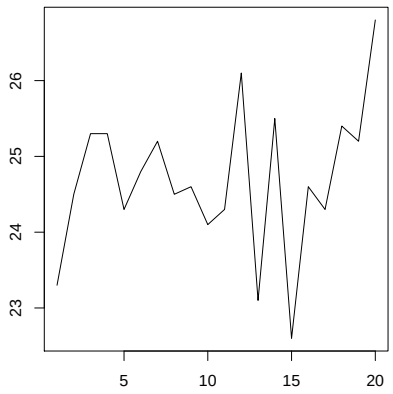
Wykres szeregu czasowego

```
> plot(cenyAkcji); # co jest równoważne:
> plot(1:20, cenyAkcji);
```



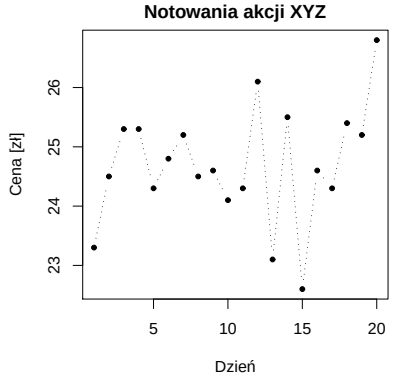
Niestety, z uzyskanego wykresu niewiele da się wyczytać. Spróbujmy połączyć punkty liniami:

```
> plot(cenyAkcji, type="l");
```



Zastosowanie funkcji kawałkami liniowej sugeruje ciągły (a w dodatku liniowy!) przyrost cen między kolejnymi punktami na osi czasu. Aby podkreślić dyskretny charakter próbkowania momentów pomiaru, można użyć parametru `type="b"` (od *both* — zarówno punkty, jak i linie).

```
> plot(cenyAkcji, type="b", pch=20, lty=3,
     main="Notowania akcji XYZ", xlab="Dzień", ylab="Cena [zł]");
```



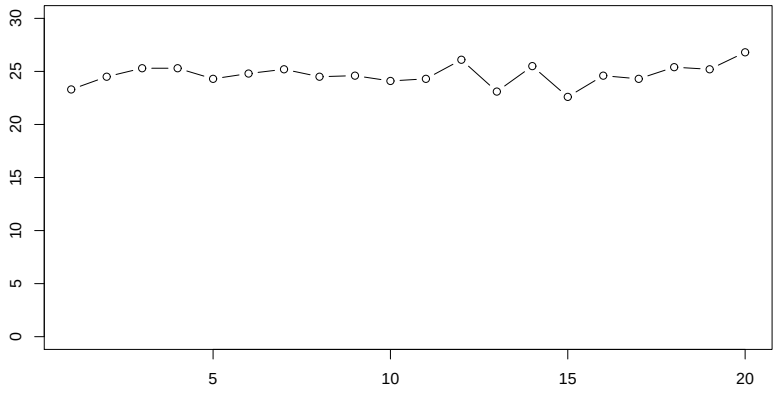
Uwaga

Zwróćmy uwagę na parametry `pch` i `lty` funkcji `plot`. Mają one za zadanie zmienić sposób rysowania symboli punktów i typów linii. Wykaz dostępnych ustawień znajduje się w *Dodatku*.

Uwaga

Warto także zanotować, jak zmienia się subiektywna percepcja zmienności cen w zależności od ustawienia zakresu danych na osi y oraz proporcji rozmiarów wykresu:

```
> plot(cenyAkcji, ylim=c(0,30), type="b");
```



□

3 Zadania do rozwiązania

Zadanie 2.6. Według danych GUS, w 2006 roku liczba urodzeń w Polsce wynosiła 374 244. Poniższa tabela zestawia te dane w zależności od wieku matki.

Wiek matki	Liczba urodzeń
19 lat i mniej	19 230
20–24	93 569
25–29	139 853
30–34	86 825
35–39	28 487
40–44	5 975
45 lat i więcej	305

a) Wprowadź dane do programu `R`.
b) Utwórz tabelę liczności i częstości liczby urodzeń w zależności od wieku matki.
c) Narysuj wykres kołowy.
d) Narysuj wykres słupkowy.
e) Zinterpretuj uzyskane wyniki.

Zadanie 2.7. Koncern paliwowy planuje otworzyć nową stację benzynową w pewnym mieście. Rozważane są cztery możliwe lokalizacje stacji — w południowej (`S`), północnej (`N`), zachodniej (`W`) i wschodniej (`E`) dzielnicy miasta. W ramach badania opinii społecznej odnośnie preferowanej lokalizacji stacji zapytano o to tysiąc kierowców. Ich odpowiedzi znajdują się w pliku `stacje.csv`. Utworzyć wykres słupkowy i wykres kołowy dla badanych preferencji.

Zadanie 2.8. Badania demograficzne przeprowadzone w 1988 roku w USA wykazały, że wśród kobiet (mających 18 i więcej lat) było: 17364 tyś. panien, 56128 tyś. mężatek, 11239 tyś. wdów i 8170 tyś. rozwódek.

a) Utwórz wykres kołowy dla stanu cywilnego danej grupy kobiet. Porównaj różne formy opisu wykresu.
b) Utwórz wykres słupkowy dla stanu cywilnego danej grupy kobiet. Porównaj różne rodzaje wykresów i formy ich opisu.

⋆ **Zadanie 2.9.** Uważa się, że oko ludzkie dobrze rozpoznaje różnice stosunków długości, lecz nie najlepiej sobie radzi ze stosunkami pól. Dlatego, w przypadku danych typu jakościowego, odradza się używania wykresu kołowego, na korzyść np. wykresu słupkowego.

a) Podaj przykład danych, dla których trudno jest ocenić, które kategorie mogą być reprezentowane liczniej od innych.
b) W dowolnym czasopiśmie poruszającym tematykę życia społeczno-politycznego (np. *Polityka, Wprost*), znajdź przykłady wykresów dla danych typu jakościowego. Których jest najwięcej?

⋆ **Zadanie 2.10.** Zanalizuj dane dotyczące liczby małżeństw w ostatnim roku według miesiąca zawarcia ślubu. Skorzystaj z aktualnego Rocznika Demograficznego publikowanego przez GUS (http://www.stat.gov.pl). Jak wyjaśnisz uzyskane wyniki?

Zadanie 2.11. Wytrzymałość na ciśnienie wewnętrzne szkła butelek jest ich ważną charakterystyką jakościową. W celu zbadania wytrzymałości butelek umieszcza się je w maszynie hydrostatycznej, po czym zwiększa się ciśnienie aż do zniszczenia butelki. Plik `butelki.csv` zawiera dane opisujące graniczną wytrzymałość na ciśnienie wewnętrzne szkła badanej partii butelek (mierzone w psi — funtach na cal kwadratowy).

a) Utwórz zmienną o nazwie `cisnienie`, opisującą wytrzymałość na ciśnienie wewnętrzne szkła butelek mierzone w MPa. Wskazówka: 1 psi = 6 894,757 Pa
b) Utwórz histogram dla danych opisujących wytrzymałość butelek. Prześledź wpływ liczby klas na kształt histogramu. Porównaj różne rodzaje histogramów.
c) Utwórz wykres łamanej liczności i nałóż go na wykres histogramu.
d) Utwórz wykres łodygowo-liściowy.
e) Utwórz i zinterpretuj wykres skrzynkowy dla wytrzymałości butelek.
f) Wyznacz i zinterpretuj podstawowe statystyki próbkowe dla danych opisujących wytrzymałość butelek.
g) Oblicz i zinterpretuj 5, 10, 25, 50, 75, 90 i 95 percentyl dla rozważanych danych.
h) Wyznacz 10% średnią uciętą dla danych opisujących wytrzymałość butelek. Porównaj średnią uciętą ze średnią arytmetyczną i medianą. Prześledzić, jak zmienia się wartość średniej wraz ze zmianą stopnia ucięcia próbki.

Zadanie 2.12. Zamieszczone poniżej dane przedstawiają wysokość czynszu płaconego w pewnej spółdzielni mieszkaniowej przez 30 losowo wybranych lokatorów.

334, 436, 425, 398, 424, 429, 392, 428, 339, 389
352, 405, 392, 403, 344, 400, 424, 443, 378, 387
384, 498, 374, 389, 367, 457, 409, 454, 345, 422.

Przeprowadź wstępną analizę statystyczną powyższych danych.

Zadanie 2.13. Przeprowadź wstępną analizę statystyczną danych dotyczących przyspieszenia (zmienna `przysp`) pojazdów z bazy `samochody.csv`, ważących mniej niż 2500 funtów (zmienna `waga`).

Zadanie 2.14. Przeprowadź wstępną analizę statystyczną danych dotyczących przyspieszenia (zmienna `przysp`) pojazdów z bazy `samochody.csv`, oddzielnie dla aut z Ameryki, Europy i Japonii.

Zadanie 2.15. Porównaj dane dotyczące mocy (zmienna `moc`) samochodów posiadających różną liczbę cylindrów (zmienna `cylindry`). Wykorzystaj informacje zawarte w bazie `samochody.csv`.

Zadanie 2.16. Pani Janina bardzo się nudzi, od czasu gdy jej pociechy założyły własne rodziny. Całe dnie spędza siedząc na ławce, obserwując życie swojej małej wioski.

Jedną z najbardziej fascynujących pozycji jej dziennego harmonogramu robót i robótek jest przybycie listonosza, pana Sławomira. Gdy przejeżdża obok płota, zdejmuje czapkę na widok staruszki, nie zsiadając z roweru. Janina dowiedziała się od naczelnika poczty, że powinien on pojawiać się u niej ok. godziny 10:25. Niestety, różnego rodzaju okoliczności zewnętrzne wpływają na fluktuację czasu przyjazdu.

Postanowiła więc zbadać „szkiełkiem i okiem" frapujący ją problem nie do końca punktualnego listonosza i zdać szczegółową sprawę jego pracodawcy. Zanotowała czasy przyjazdów (w minutach po godz. 10-tej) w kolejnych 33 dniach roboczych. Potem jednak okazało się, że 3 wartości są nieczytelne z powodu pisania nienaostrzonym ołówkiem.

26, 22, 26, 20, 25, ??, 21, 20, 28, 27, 26,
38, 23, 30, 21, 25, 26, 23, 25, 27, 27, ??,
25, 22, 23, 31, 19, ??, 25, 25, 23, 25, 24.

Po wsi swego czasu krążyły plotki o wyższości `R`-a nad innymi programami w swojej klasie. Poprosiła więc Ciebie, wielce pilnego studenta, o pomoc.

Przeprowadź wstępną analizę tego zbioru używając wszystkich znanych Ci sposobów.

Zadanie 2.17. Z danych z poprzedniego zadania usuń obserwacje odstające i braki danych. Następnie przyporządkuj każdej obserwacji jedną z pięciu kategorii:

Kategoria	Czas przyjazdu
ZaWcz	$(-\infty, 23)$,
Wcz	$[23, 25)$,
Punkt	$= 25$,
Pźn	$(25, 27]$,
ZaPźn	$(27, \infty)$.

Opisz wynikowy zbiór za pomocą znanych Ci metod.

⋆ **Zadanie 2.18.** W poprzednim dziale kilka wykresów zostało tak narysowanych, aby wywołać różnego rodzaju subiektywne efekty na odbiorcy (np. zmniejszenie wrażenia wielkości zmienności). Znajdź inne przykłady manipulacji spotykane w życiu codziennym, np. w czasopismach bądź w materiałach reklamowych.

⋆ **Zadanie 2.19.** Napisz funkcję wyznaczającą dla danej realizacji próby $\mathbf{X} = (X_1, \ldots, X_n)$ i.i.d. wartość nieobciążonego estymatora kurtozy $\kappa_{\mathbf{X}}$:

$$\kappa_{\mathbf{X}} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{\mathbf{X}}}{s_{\mathbf{X}}} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}, \tag{2}$$

gdzie, $\overline{\mathbf{X}}$ — średnia z próby, $s_{\mathbf{X}}$ — odchylenie standardowe.

⋆ **Zadanie 2.20.** Napisz funkcję wyznaczającej dla danej realizacji próby wartość średniej winsoryzowanej dowolnego rzędu.

4 Wskazówki

Wskazówka do zadania 2.16. Rozważ nie tylko wymienione we Wprowadzeniu metody dla danych jakościowych, ale także potraktuj wyniki pomiarów jako szereg czasowy.

Wskazówka do zadania 2.19. Tworzenie funkcji w `R` dokonuje się wg następującej składni:

```
> nazwaFunkcji <- function(argument1, argument2, (...))
{
   (...) różne operacje (...)
   return(wynik);
}
```

Na przykład funkcja licząca średnią arytmetyczną może być utworzona za pomocą operacji:

```
> srednia <- function(probka)
  {
     suma <- sum(probka);
     licznosc <- length(probka);
     return(suma/licznosc);
  }
```

Sprawdźmy:

```
> X <- c(1,2,3,4,5,6);
> srednia(X);
[1] 3.5
```

Bibliografia

[1] Adam Grobler. *Metodologia nauk.* Znak, Warszawa, 2006.

[2] Przemysław Grzegorzewski, Konstancja Bobecka, Anna Dembińska, Jerzy Pusz. *Rachunek prawdopodobieństwa i statystyka.* Wyd. WSISiZ, Warszawa, 2008.

[3] Jacek Koronacki, Jan Mielniczuk. *Statystyka.* WNT, Warszawa, 2001.